

# Implementasi *Nearest Neighbor* pada Data Kategorik dengan Pembobotan Atribut Menggunakan *Weighted Simple Matching Coefficient*

Romario Wijaya, Nanik Suciati, dan Wijayanti Nurul Khotimah

Departemen Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: nanik@if.its.ac.id

**Abstrak**—Klasifikasi merupakan sebuah metode yang cukup sering digunakan dalam bidang statistika dan komputasi cerdas terutama dengan kaitannya untuk memprediksi sesuatu berdasarkan data yang diterima. Klasifikasi dapat dilakukan untuk berbagai macam tipe data mulai dari numerik, ordinal ataupun kategorik. Pada data kategorik perhitungan jarak perbedaan atau kesamaan tidak dapat dihitung menggunakan operasi matematik seperti yang dapat dilakukan terhadap data numerik. Salah satu metode yang dapat digunakan untuk menghitung jarak pada data kategorik adalah *Simple Matching Coefficient (SMC)*. Namun metode ini terkesan tidak adil karena menganggap bahwa setiap atribut memiliki kontribusi yang sama terhadap tiap label kelas. Karena itu metode SMC akan dioptimasi dengan menggunakan *Weighted Simple Matching Coefficient (WSMC)*. Dalam penelitian kali ini, diimplementasikan sebuah metode klasifikasi terhadap data kategorik yang menggunakan metode klasifikasi *Nearest-Neighbor* dengan pembobotan atribut menggunakan WSMC. Uji coba yang dilakukan terhadap 6 dataset dengan atribut bersifat kategorik, menunjukkan kemampuan metode dalam melakukan klasifikasi. Dengan rata-rata akurasi tertinggi didominasi oleh metode lokal. Rata-rata akurasi tertinggi untuk dataset *Nursery*, *Cars*, Gerakan tangan, *Soybeans*, *Vote* dan *Dermatology* berturut-turut sebesar 77.31%, 80.35%, 97.00%, 91.64%, 92.41%, dan 95.90%.

**Kata Kunci**—*Nearest-Neighbor*, data kategorik, pembobotan atribut.

## I. PENDAHULUAN

PADA era saat ini, aplikasi komputer atau perangkat bergerak yang diperuntukkan untuk para pakar maupun yang diperuntukkan untuk orang awam terus dikembangkan ke arah yang lebih cerdas untuk memudahkan pekerjaan manusia. Aplikasi yang dibuat haruslah memiliki kemampuan untuk memberikan rekomendasi hasil analisa input. Contoh input yang paling sering dianalisa adalah data, baik itu data numerik maupun data kategorik. Di sinilah peran klasifikasi sangat dibutuhkan oleh para pembuat aplikasi untuk terus mengembangkan aplikasinya agar semakin cerdas dan diminati banyak orang.

Metode klasifikasi yang paling umum, simpel dan mudah untuk diterapkan salah satunya adalah *k-Nearest-Neighbor (k-NN)*. Algoritma *k-NN* dinilai cepat karena merupakan *Instance-based learning* atau *lazy-learning* [1] yang tidak memerlukan fase latihan pada awal proses [2]. Namun dalam penerapannya, untuk melakukan klasifikasi terhadap data kategorik, diperlukan cara khusus dalam menghitung jarak

antar data yang ada. Salah satu cara khusus dalam menghitung jarak antar data yang paling sering diterapkan adalah menggunakan *Simple Matching Coefficient (SMC)*[3].

Dalam penelitian kali ini, diimplementasikan sebuah metode klasifikasi terhadap data kategorik yang menggunakan metode klasifikasi *Nearest-Neighbor* dengan pembobotan atribut menggunakan *Weighted Simple Matching Coefficient (WSMC)*.

Diskusi pada jurnal ini dibagi dalam struktur sebagai berikut: Bab II membahas materi dan metode yang digunakan dalam sistem. Bab III membahas hasil dan diskusi uji coba yang dilakukan terhadap enam dataset dengan atribut data kategorik. Terakhir pada Bab IV membahas kesimpulan dan saran dari hasil yang didapatkan dalam penelitian ini.

## II. MATERI DAN METODE

Pada sistem yang dibangun, terdapat dua tahap penting yang dilakukan. Dua tahap tersebut adalah tahap *training* dan tahap *testing*. Tahap *training* adalah tahap dimana proses perhitungan bobot atribut dilakukan, dan tahap *testing* adalah tahap dimana proses klasifikasi *k-NN* dilakukan dengan pembobotan atribut yang sudah didapatkan pada tahap *training*. Sistem menggunakan metode *Weighted Simple Matching Coefficient* untuk melakukan perhitungan jarak antar data dan *k-Nearest Neighbor* untuk proses klasifikasi.

### A. *Weighted Simple Matching Coefficient (WSMC)*

*Weighted Simple Matching Coefficient (WSMC)* adalah perhitungan statistik yang hampir sama dengan *Simple Matching Coefficient* yang melakukan perbandingan untuk tiap atribut dengan atribut yang lain, namun dalam implementasinya, setiap atribut tidak akan dihitung secara rata, namun dengan memberikan bobot terhadap tiap atribut sehingga secara tidak langsung mengurangi pengaruh data yang akan mengacaukan klasifikasi. Metode perhitungan ini dapat digunakan secara efektif untuk melakukan klasifikasi terhadap data yang bersifat kategorik[3]. Pengukuran jarak baru akan dinotasikan dengan  $WSMC_{global}$  dan  $WSMC_{local}$  tergantung dari metode pembobotan(global atau lokal) yang digunakan. Pada metode global, atribut dikaitkan dengan vektor bobot  $\omega = (\omega_1, \dots, \omega_d, \dots, \omega_D)$  dan jarak WSMC antara objek data  $x_i$  dan  $x_j$  diberikan pada persamaan

(1).

$$WSMC_{global}(x_i, x_j, \omega) = \sum_{d=1}^D \omega_d \times I(x_{id} \neq x_{jd}) \quad (1)$$

Dimana  $I(.)$  adalah fungsi indikator dengan  $I(true) = 1$  dan  $I(false) = 0$ , dan  $\omega_d$  adalah bobot unik yang diberikan kepada atribut  $d$  yang akan dihitung berdasarkan dari *Global Entropy* ( $GE$ ) dan *Global Gini diversity index* ( $GG$ ). Dari perspektif peringkat fitur, nilai *Entropy* dan *Gini diversity index* di sini mengukur derajat kontribusi dari atribut dalam mendiskriminasikan kelas *training*. Secara formal, *Entropy* dan *Gini diversity index* dapat dihitung dengan persamaan (3) dan (5).

Pada metode lokal, tujuannya adalah memberikan bobot yang didapat berdasarkan kelas untuk tiap atribut, yang mengindikasikan kontribusi yang berbeda oleh tiap atribut terhadap kelas yang berbeda. Pada kasus ini, kelas ke- $m$  dikaitkan dengan vektor bobot  $w_m = (w_{m1}, \dots, w_{md}, \dots, w_{mD})$  dengan  $0 \leq w_{md} \leq 1$  untuk  $d = 1, 2, \dots, D$ . Jarak SMC antara objek data  $x_i$  dan  $x_j$  berubah menurut persamaan (2).

$$WSMC_{local}(x_i, x_j, w_m) = \sum_{d=1}^D w_{md} \times I(x_{id} \neq x_{jd}) \quad (2)$$

Bobot lokal dapat dihitung juga menggunakan informasi *Entropy* dan *Gini diversity index*, seperti pada bobot global, namun dengan menggunakan *Local Entropy* ( $LE$ ) dan *Local Gini diversity index* ( $LG$ ). Adapun  $LE$  dan  $LG$  didefinisikan dalam persamaan (9) dan (11).

$$GE(s_d) = - \sum_{m=1}^M p(m|s_d) \log_2 p(m|s_d) \quad (3)$$

Dengan  $s_d$  menotasikan sebuah kategori ke- $d$ ,  $M$  adalah jumlah kelas yang ada pada dataset dan  $p(m|s_d)$  didefinisikan dengan persamaan (4).

$$p(m|s_d) = \frac{\sum_{(x,y) \in c_m} I(x_d = s_d)}{\sum_{(x,y) \in tr} I(x_d = s_d)} \quad (4)$$

Pada persamaan (4) dilakukan perhitungan terhadap tiap data yang pada kelas  $m$  yang memiliki atribut ke  $d$  bernilai  $s_d$  dan dibagi dengan jumlah semua data yang memiliki atribut ke  $d$  bernilai  $s_d$ . Persamaan (3) disebut *Global Entropy* karena dihitung menggunakan  $p(m|s_d)$  yang menyebabkan nilainya unik terhadap tiap kelas yang ada  $m$ . Hal ini menyebabkan persamaan (5) juga disebut sebagai *Global Gini diversity index* mengingat perhitungan juga menggunakan  $p(m|s_d)$ .

$$GG(s_d) = 1 - \sum_{m=0}^M [p(m|s_d)]^2 \quad (5)$$

Dengan  $p(m|s_d)$  dihitung dengan persamaan (4). Perhitungan persamaan (3) dan (5) berpengaruh besar terhadap perhitungan *Global Weight* ( $\omega_d$ ). Apabila  $\omega_d$  dihitung menggunakan  $GE$  akan menghasilkan  $\omega_d^{(GE)}$  yang dapat dihitung menggunakan persamaan (6), sedangkan apabila  $\omega_d$  dihitung menggunakan  $GG$  akan menghasilkan  $\omega_d^{(GG)}$  yang dapat dihitung menggunakan persamaan (8).

Pada persamaan (6) dan (8), muncul sebuah variabel baru  $0 \leq p(s_d) \leq 1$  yang merupakan derajat pengaruh atribut  $s_d$  terhadap semua data yang digunakan. Adapun  $p(s_d)$  dapat

dihitung menggunakan persamaan (7).

$$\omega_d^{(GE)} = e^{\frac{1}{\log_2 M} \sum_{s_d \in S_d} p(s_d) \times GE(s_d)} \quad (6)$$

$$p(s_d) = \frac{1}{N} \sum_{(x,y) \in tr} I(x_d = s_d) \quad (7)$$

Dengan  $tr$  adalah *training dataset* dan  $N$  menunjukkan jumlah data.

$$\omega_d^{(GG)} = e^{\frac{M}{M-1} \sum_{s_d \in S_d} p(s_d) \times GG(s_d)} \quad (8)$$

Sama halnya dengan perhitungan pada  $GE$  dan  $GG$ , perhitungan  $LE$  dan  $LG$  pada persamaan (9) dan (11) tidak jauh berbeda. Yang memberikan perbedaan pada metode Global dan Lokal adalah pada perhitungan derajat kontribusi tiap atribut terhadap tiap kelas. Pada metode Global, digunakan  $p(m|s_d)$  yang perhitungannya melibatkan semua data yang ada pada data *training*. Berbeda dengan metode Global, pada metode Lokal dihitung menggunakan  $p(s_d|m)$  yang tidak melibatkan semua data yang ada pada data *training* namun hanya menghitung spesifik terhadap satu kelas khusus saja.

$$LE(m, d) = - \sum_{s_d \in S_d} p(s_d|m) \log_2 p(s_d|m) \quad (9)$$

Dengan  $p(s_d|m)$  didefinisikan dengan persamaan (10).

$$p(s_d|m) = \frac{1}{|C_m|} \sum_{(x,y) \in C_m} I(x_d = s_d) \quad (10)$$

Dengan  $C_m$  merupakan kelas ke- $m$  dan  $|C_m|$  adalah jumlah sampel pada kelas ke- $m$ .

$$LG(m, d) = 1 - \sum_{s_d \in S_d} [p(s_d|m)]^2 \quad (11)$$

Dengan  $p(s_d|m)$  dihitung menggunakan persamaan (10). Berbeda dengan pembobotan global, bobot lokal yang masing-masing dihitung dengan persamaan (12) dan (13) tidak menggabungkan nilai dari tiap-tiap  $LE$  dan  $LG$  yang sudah dihitung seperti yang dilakukan pada metode global. Pada pembobotan lokal, bobot akan dihitung secara berulang kali terhadap tiap  $LE$  dan  $LG$ , sehingga akan didapatkan matriks bobot yang ukurannya lebih besar, namun menunjukkan hubungan antara sebuah atribut dengan masing-masing kategori label kelas secara spesifik dan unik. Contoh bobot atribut yang didapatkan dapat dilihat pada Tabel 1.

$$w_{md}^{(LE)} = e^{\frac{1}{\log_2 |S_d|} \times LE(m, d)} \quad (12)$$

$$w_{md}^{(LG)} = e^{\frac{|S_d|}{|S_d|-1} \times LG(m, d)} \quad (13)$$

Tabel 1.  
Contoh Tabel Bobot yang Didapatkan

Attribu	Attribu	Attribu	Attribu	Attribu	Attribu
---------	---------	---------	---------	---------	---------

	te #1	te #2	te #3	te #4	te #5	te #6
WeightG E	0.5831	0.5685	0.5481	0.6075	0.5560	0.6225
WeightG G	0.5584	0.5526	0.5441	0.5946	0.5487	0.6019
WeightL EC1	0.3808	0.3704	0.3687	0.5323	0.3753	0.5381
WeightL EC2	0.6371	0.6271	0.3688	0.5336	0.3713	0.5396
WeightL EC3	0.3825	0.3729	0.3685	0.3801	0.3692	0.3824
WeightL EC4	0.6301	0.4651	0.3841	0.5338	0.5441	1
WeightL GC1	0.3793	0.3702	0.3686	0.4727	0.3757	0.4811
WeightL GC2	0.5611	0.5455	0.3687	0.4746	0.3715	0.4833
WeightL GC3	0.3799	0.3727	0.3684	0.3820	0.3694	0.3842
WeightL GC4	0.5502	0.4247	0.3821	0.4749	0.4899	1

### B. *k*-Nearest Neighbor dengan WSMC

*k*-Nearest Neighbor dengan WSMC adalah sebuah metode pengembangan dari *k*-NN dengan SMC, dimana di dalam *k*-Nearest Neighbor dengan SMC, atribut yang ada dihitung begitu saja dengan bobot yang sama rata. Dalam pengembangan metode ini, perhitungan terhadap jarak antar atribut tetap menggunakan SMC, namun dengan tambahan pembobotan untuk tiap atributnya, sehingga tiap atribut memiliki bobot peran yang berbeda dalam menentukan hasil klasifikasi (lihat persamaan (1) dan (2)). Algoritma *learning* dapat dilihat pada **Gambar 1. Algoritma 1** dan **Gambar 2. Algoritma 2**.

Pada **Gambar 1. Algoritma 1** bobot dari tiap atribut akan dihitung menggunakan metode global, sehingga bobot yang dipakai adalah  $\omega_d^{(GE)}$  (jika dipilih fitur *Entropy*) atau  $\omega_d^{(GG)}$  (jika dipilih fitur *Gini Diversity index*). Bobot  $\omega_d$  yang telah digantikan oleh  $\omega_d^{(GE)}$  atau  $\omega_d^{(GG)}$  selanjutnya akan digunakan untuk melakukan klasifikasi dengan cara mengalikan bobot atribut dengan jarak yang sudah dihitung menggunakan SMC untuk tiap atributnya.

Sedangkan untuk **Gambar 2. Algoritma 2**, bobot dari tiap atribut akan dihitung menggunakan metode lokal, sehingga bobot yang dipakai adalah  $w_{md}^{(LE)}$  (jika dipilih fitur *Entropy*) atau  $w_{md}^{(LG)}$  (jika dipilih fitur *Gini Diversity index*). Berbeda dengan metode global, pada metode lokal bobot yang terbentuk tidak hanya satu untuk tiap atribut, namun sejumlah label kelas yang ada. Hal ini dimaksudkan agar nantinya ketika data diklasifikasikan, bobot atribut yang dipakai adalah bobot atribut yang sesuai dengan karakter label kelas tertentu. Setelah bobot lokal terbentuk, dilakukan proses klasifikasi dengan cara mengalikan jarak yang dihitung menggunakan SMC dengan bobot lokal untuk masing-masing label kelas. Adapun secara rinci, *pseudocode* *k*-Nearest Neighbor dengan WSMC dapat dilihat pada **Gambar 3. Algoritma 3**.

**Input:**  $tr = \bigcup_{m=1}^M c_m$ , dan GE atau GG yang ditentukan untuk metode pembobotan yang berbeda.  
**Output:**  $\omega = \langle \omega_1, \dots, \omega_d, \dots, \omega_D \rangle$   
**begin**  
  **for**  $d=1$  to  $D$  **do**

**if** GE yang dipilih **then**  
   $\omega_d \leftarrow \omega_d^{(GE)}$  menggunakan (6)  
**else**  
   $\omega_d \leftarrow \omega_d^{(GG)}$  menggunakan (8)  
**end**  
**end**

Gambar 1. Algoritma 1 *Global Weights Learning Algorithm* GWLA

**Input:**  $c_m$ , dan LE or LG yang ditentukan untuk metode pembobotan yang berbeda.  
**Output:**  $W_m = \langle W_{m1}, \dots, W_{md}, \dots, W_{mD} \rangle$   
**begin**  
  **for**  $d=1$  to  $D$  **do**  
    **if** LE yang dipilih **then**  
       $\omega_{md} \leftarrow w_{md}^{(LE)}$  menggunakan (12)  
    **else**  
       $\omega_{md} \leftarrow w_{md}^{(LG)}$  menggunakan (13)  
  **end**  
  **for**  $d=1$  to  $D$  **do**  
     $W_{md} \leftarrow \frac{\omega_{md}}{\sum_{d=1}^D \omega_{md}}$ , sedemikian hingga  $\|w_m\|_1 = 1$ .  
  **end**  
**end**

Gambar 2. Algoritma 2 *Local Weights Learning Algorithm* LWLA

**Input :**  $tr$ , konstanta nearest neighbors  $k$ , dan sampel tes  $z = (x, y)$   
**Output :**  $y$ , label kelas dari  $x$   
**Begin**  
  hitung WSMC  $(x, x_i, w)$ , jarak antara  $z$  dengan setiap sampel  $(x_i, y_i) \in tr$ ;  
  ilih  $NN_z \subseteq tr$ ,  $k$  set dari  $k$  sampel *training* terdekat terhadap  $z$ , dari segi jarak;  
  **Output**  $y = \arg\max_m \sum_{(x_j, y_j) \in NN_z} I(m = y_j)$   
**End**

Gambar 3. Algoritma 3 *k*-NN dengan WSMC

### III. HASIL DAN DISKUSI UJI COBA

Data uji coba yang digunakan sebagai masukan adalah data tabel dalam format .csv. Adapun isi dari tabel adalah atribut yang bersifat kategorik, diikuti dengan label kelas pada kolom terakhir. Jumlah data, jumlah atribut, jumlah kategori atribut dan jumlah kategori label kelas pada dataset uji coba bervariasi untuk melakukan uji coba karakteristik masing-masing metode pembobotan.

Untuk data yang digunakan serta parameter jumlah atribut dan jumlah kategori label kelas dapat dilihat pada Tabel 2. Adapun untuk dataset selain data gerakan tangan didapatkan dari <http://ftp.ics.uci.edu/pub/machine-learning-databases/>. Pengujian ditujukan untuk mendapatkan tingkat akurasi rata-rata untuk tiap dataset. Adapun pengujian dilakukan dengan metode *k*-Fold Cross Validation. Dengan menggunakan metode *k*-Fold Cross Validation hasil yang didapatkan dari pengujian dapat lebih merata dan adil untuk tiap datasetnya.

Uji coba juga dilakukan terhadap metode k-NN dengan SMC tanpa pembobotan atribut yang diimplementasikan pada MATLAB dan WEKA.

Hasil uji coba ditampilkan dalam bentuk tabel dengan masing-masing akurasi untuk tiap dataset dan tiap metode. Akurasi paling tinggi ditunjukkan dengan tanda bintang(\*) pada dan dapat dilihat pada Tabel 3.

Tabel 2.

Data yang Digunakan untuk Uji Coba

Data Set	#Atribut (D)	Contoh nilai atribut	#Kelas (M)	#Jumlah Data (N)
<i>Nursery</i>	8	<i>Recommended, priority, not_recom</i>	5	12958
<i>Cars</i>	6	<i>Vgood, good, 5more</i>	4	1728
Gerakan Tangan	36	6, 7, 8 (kategorik)	10	150
<i>Soybeans</i>	35	3, 0, 2 (kategorik)	19	683
<i>Vote</i>	16	Y, N	2	435
<i>Dermatology</i>	34	0, 1, 2 (kategorik)	6	366

Tabel 3.  
Hasil Uji Coba Akurasi (%)

Dataset	GE-NN	GG-NN	LE-NN	LG-NN	k-NN SMC (Matlab)	k-NN (WEKA)
<i>Nursery</i>	72.45	72.76	76.92	77.31*	40.04	76.38
<i>Cars</i>	77.46	77.46	80.35*	79.77	64.74	74.77
Gerakan Tangan	90.50	90.50	93.50	97.00*	92.00	95.50
<i>Soybeans</i>	89.31	89.01	91.65*	90.62	88.72	91.22
<i>Vote</i>	91.95	92.18	92.41*	91.03	91.26	91.95
<i>Dermatology</i>	92.34	91.80	95.90*	95.90*	90.43	95.35

Untuk setiap data, dari hasil uji coba yang sudah dilakukan, rata-rata metode lokal yang memiliki tingkat akurasi rata-rata yang tinggi. Jika dibandingkan dengan metode lokal, metode global memiliki tingkat akurasi yang sedikit lebih rendah. Namun, Dari hasil uji coba tersebut, dapat dilihat bahwa metode pembobotan atribut rata-rata memiliki tingkat akurasi lebih baik dibandingkan dengan k-NN tanpa pembobotan atribut.

Metode lokal dapat memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan metode global, dikarenakan sifat dari data uji coba itu sendiri. Dalam klasifikasi, data yang diberikan adalah data sebenarnya dari sebuah eksperimen atau uji coba. Yang mana setiap label kelas sudah memiliki ciri tersendiri, hal ini menyebabkan persebaran data seakan-akan telah memiliki kelompok-kelompok tersendiri. Hal ini menyebabkan metode lokal lebih unggul mengingat metode lokal memberi bobot spesifik untuk tiap label kelas yang berbeda. Hal ini dibuktikan dari beberapa data uji coba.

Seperti pada data *Nursery* dan *Cars*, metode global cenderung memiliki tingkat akurasi yang berselisih jauh

dengan metode lokal. Ini karena data *Nursery* dan *Cars* memiliki jumlah data yang cukup banyak namun dengan jumlah label kelas yang sedikit. Berbeda dengan performa untuk data *Soybeans*, yang memiliki jumlah kelas cukup banyak namun jumlah data yang sangat sedikit, metode global dapat memperkecil selisih ketertinggalannya hingga hanya sekitar 2%. Hal ini dapat memberikan sedikit gambaran tentang bagaimana karakteristik tiap metode baik itu lokal maupun global.

#### IV. KESIMPULAN DAN SARAN

Metode klasifikasi k-NN dapat lebih dimaksimalkan dengan menggunakan pembobotan atribut baik itu global maupun lokal. Pemberian bobot kepada k-NN dapat melalui beberapa metode perhitungan yaitu *Global Entropy*, *Global Gini Diversity*, *Local Entropy*, dan *Local Gini Diversity*. Berdasarkan uji coba yang telah dilakukan, apabila diamati lebih lanjut, pemilihan metode perhitungan bobot menggunakan metode global atau lokal harus mempertimbangkan bentuk persebaran data. Metode global kemungkinan lebih cocok untuk data berjumlah sedikit namun memiliki label kelas yang cukup banyak sehingga tidak ada data yang seakan-akan telah memiliki kelompok tersendiri seperti data *Soybeans* atau data lain yang memiliki jumlah label kelas lebih banyak, sedangkan metode lokal lebih cocok digunakan pada data yang berjumlah banyak dengan label kelas sedikit sehingga data cenderung berkelompok dengan karakteristik tiap label kelas seperti pada data *Nursery*, *Cars*, dan Gerakan Tangan yang telah diuji cobakan sebelumnya.

Performa metode k-NN dengan pembobotan atribut dapat dikatakan lebih baik di dalam hal akurasi dibandingkan dengan metode k-NN tanpa pembobotan atribut. Berdasarkan uji coba yang telah dilakukan terhadap beberapa dataset sebelumnya, terbukti metode k-NN dengan pembobotan atribut memiliki akurasi yang lebih tinggi dibandingkan dengan metode k-NN normal yang dilakukan tanpa pembobotan atribut.

Untuk penelitian selanjutnya, terdapat beberapa hal yang dapat diperbaiki. Pertama, perbaikan dataset sebelum dilakukan *training* maupun *testing* untuk menghilangkan *missing value* atau reduksi dimensi dan menghilangkan *noise* (*Preprocessing*). Kedua, pemaksimalan metode klasifikasi dengan menggunakan metode perhitungan jarak yang lainnya. Ketiga, penggunaan metode klasifikasi yang lain, mengingat kompleksitas dari metode klasifikasi k-NN adalah cukup besar yaitu  $O(n^2)$ .

#### DAFTAR PUSTAKA

- [1] "k-nearest neighbors algorithm," *Wikipedia*. Nov-2016.
- [2] Wikipedia, "Instance-based learning," *Wikipedia*, 2016.
- [3] L. Chen and G. Guo, "Nearest neighbor classification of categorical data by attributes weighting," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3142–3149, Apr. 2015.